IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, May 2026
Impact Factor: 8.2

PyqSpot: An Intelligent Exam Preparation Platform Using Dynamic Mock Test Generation

Pisal Omkar Kisan[1], Satalkar Vinay Balu[2], Pawar Avadhoot Subhash[3], Thakare Samradnya Sanjay[4], Prof. Dhas Renuka Premraj[5]
Final Year Student, Department of Computer Engineering[1,2,3,4]
Professor, Department of Computer Engineering[5]
Hon Shri Babanrao Pachpute Vichardhara Trusts, GOI Faculty of Engineering Kashti, Ahmednagar, India

Abstract: *Engineering students at Savitribai Phule Pune University (SPPU) face significant challenges in accessing verified, organized and interactive study resources for semester examination preparation. Existing solutions rely on fragmented, unverified resources such as Telegram groups and shared drives that lack centralized management, secure delivery and AI-driven interactivity.*

This paper presents PYQSPOT, an intelligent full-stack web platform developed using Next.js, Node.js and Express.js that addresses this gap by providing a verified digital repository of solved Previous Year Question Papers (PYQs) with a secure, automated Razorpay payment gateway and PDF delivery pipeline through the Google Drive API. The system introduces a Dynamic Mock Test Generator powered by the Google Gemini 2.5 Flash API, which analyses uploaded question paper text and generates randomized multiple-choice questions in real time, enabling active self-assessment with up to three attempts per paper. PYQSPOT implements a three-tier role-based architecture encompassing students, admins and a super admin, with a comprehensive analytics dashboard for revenue tracking, login statistics and performance monitoring.

The platform is deployed on Hostinger with MySQL as the relational database and supports dynamic hierarchy management for universities, branches, semesters, subjects and exam patterns across SPPU Engineering programs. Currently serving over 2,000 active students with 100+ verified PDF solutions, PYQSPOT demonstrates the commercial viability and academic impact of integrating AI-driven assessment with secure digital content delivery.

Keywords: PYQSPOT, Dynamic Mock Test Generation, E-Learning, Previous Year Questions (PYQs), Artificial Intelligence, Adaptive Learning, Educational Technology, Next.js, SPPU.

I. INTRODUCTION

The rapid expansion of digital education has transformed how students access and interact with academic resources. Engineering students at Savitribai Phule Pune University (SPPU) are particularly affected by inefficiencies in the existing academic ecosystem, where examination preparation resources are scattered, unverified and inaccessible through a unified platform. Research indicates that approximately 60-70% of semester examination questions are repeated or conceptually derived from Previous Year Question Papers (PYQs), making access to well-organized, verified PYQ resources critical for effective examination preparation [1].

PYQSPOT was conceived to eliminate these inefficiencies by providing a centralized, intelligent and commercially viable EdTech platform. The system integrates a hierarchical content management system for organizing PYQs across universities, branches, semesters and subjects, a secure payment and automated PDF delivery pipeline, and an AI-powered Dynamic Mock Test Generator that creates personalized, randomized multiple-choice assessments from uploaded question paper content.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, May 2026 Impact Factor: 8.2

The platform architecture follows a client-server model with Next.js 14 as the frontend framework, Express.js with Node.js as the backend API server and MySQL as the relational database. External integrations include the Razorpay Payment Gateway for secure transactions, the Google Drive API for encrypted PDF storage and delivery, and the Google Gemini 2.5 Flash API for natural language processing and intelligent question generation.

This paper presents the complete system design, implementation methodology, AI integration workflow and results of the PYQSPOT platform, demonstrating how modern web technologies and artificial intelligence can be combined to create a scalable, revenue-generating educational tool that transitions students from passive reading to active, AI-driven self-assessment.

II. PROBLEM STATEMENT

Engineering students often face difficulties in accessing reliable and organized study resources for university examination preparation. Most students depend on fragmented platforms such as Telegram groups, shared drives and unverified websites for Previous Year Questions (PYQs) and solved papers. These resources are frequently incomplete, outdated and difficult to manage.

Traditional PDF-based learning systems only support passive learning and lack interactive features such as intelligent mock tests, automated evaluation and performance tracking. Students spend significant time searching for study materials instead of focusing on effective preparation and self-assessment.

Additionally, existing systems generally do not provide secure digital delivery, centralized management or AI-based adaptive learning functionalities. Therefore, there is a need for an intelligent and scalable exam preparation platform that combines verified PYQ resources, secure cloud integration and AI-based Dynamic Mock Test Generation to improve examination readiness and learning efficiency

III. LITERATURE REVIEW

Several research studies and educational platforms have contributed to the development of intelligent e-learning and online examination systems.

Romi (2023) proposed an adaptive e-learning system success model that improves learner performance through interactive digital learning environments and adaptive educational techniques [1]. Ali (2023) discussed secure and scalable cloud-based e-commerce architectures for digital platforms, highlighting the importance of scalability, security, cloud storage and performance optimization for online systems handling digital educational resources [2].

Deng et al. (2024) developed an innovative Question Bank Management System (QBMS) using Vue.js and Spring Boot for efficient management of educational question banks and personalized learning support [3]. Alshboul et al. (2024) presented a hybrid approach for Automatic Question Generation (AQG) using Artificial Intelligence and semantic analysis techniques to generate intelligent questions for e-learning platforms [4]. Prasanth et al. (2024) proposed a mock test paper system that provides online practice tests, instant evaluation and self-assessment features for students preparing for examinations [5].

Although these systems provide valuable contributions in e-learning, cloud architecture, question generation and online testing, most of them do not focus on centralized PYQ-based engineering exam preparation with AI-driven dynamic mock test generation. PYQSPOT aims to bridge this gap by integrating verified PYQ resources, intelligent mock test generation and interactive self-assessment into a single commercially deployed platform.

IV. EXISTING SYSTEM

Existing educational platforms mainly provide static learning materials such as PDFs, notes and recorded lectures. Most engineering students rely on Telegram groups, Google Drive folders and unofficial websites to access Previous Year Questions (PYQs) and solutions. These platforms are usually unorganized, lack proper verification and do not provide centralized management.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, May 2026 Impact Factor: 8.2

Traditional systems primarily support passive learning where students only read or download content without active self-assessment. Most existing systems also lack secure payment integration, cloud-based resource delivery and intelligent functionalities such as AI-driven mock test generation and performance analytics.

Furthermore, many systems fail to provide personalized learning experiences and scalable educational services. This creates inefficiency in exam preparation and reduces student engagement. Therefore, there is a need for a modern and intelligent platform that integrates verified PYQ resources with AI-based testing, cloud storage and interactive learning features. Table I presents a comparative analysis of existing approaches against PYQSPOT.

TABLE I

Feature	Telegram/WhatsApp	Google Drive	Generic Quiz Tools	PYQSPOT
Verified Content	No	No	N/A	Yes
Centralized Access	No	No	No	Yes
Secure Payment	No	No	No	Yes
AI Mock Tests	No	No	No	Yes
Performance Analytics	No	No	No	Yes
Admin Dashboard	No	No	No	Yes
Role-Based Access	No	No	No	Yes

V. PROPOSED METHODOLOGY

PYQSPOT follows an agile, iterative development methodology organized into four functional phases.

Phase 1 — Content Repository & Authentication: A hierarchical MySQL database structure was designed to model the academic hierarchy of University → Exam Pattern → Branch → Semester → Subject → Paper. JWT-based authentication with bcrypt password hashing provides secure role-based access for students, admins and the super admin.

Phase 2 — Secure Commerce Pipeline: The Razorpay Payment Gateway was integrated to handle UPI, card and net banking transactions. Upon successful payment verification through HMAC-SHA256 cryptographic signature validation, the Express.js backend dynamically generates a time-limited download link via the Google Drive API and delivers it to the student. An automated email receipt system using Nodemailer with IST timestamps ensures transaction transparency.

Phase 3 — AI-Powered Mock Test Engine: The Google Gemini 2.5 Flash API receives structured prompts containing uploaded question paper text and returns a validated JSON array of 10 MCQs with four options and correct answers. Questions are shuffled using the Fisher-Yates algorithm for each attempt, with a maximum of three attempts per paper to maintain assessment integrity.

Phase 4 — Analytics & Administration: A comprehensive admin dashboard provides real-time revenue analytics, user login statistics, support ticket management and full hierarchy management. A super admin can dynamically add universities, branches, semesters and subjects without code changes, ensuring the platform scales to new academic structures.

VI. SYSTEM ARCHITECTURE

PYQSPOT implements a three-layer client-server architecture as described below.

A. Presentation Layer (Frontend): The frontend is built with Next.js 14 utilizing the App Router and Server-Side Rendering (SSR) for improved SEO and load performance. Tailwind CSS provides a responsive, mobile-first user interface accessible on all devices. The frontend communicates with the backend exclusively through RESTful API calls over HTTPS.

B. Application Layer (Backend): The Express.js and Node.js backend serves as a secure API controller managing authentication (JWT), payment processing (Razorpay webhooks), file delivery (Google Drive API) and AI integration

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, May 2026
Impact Factor: 8.2

(Gemini API). Role-based middleware enforces access control at the route level, distinguishing between student, admin and superadmin permissions.

C. Data Layer (Database): MySQL is used as the primary relational database with Sequelize ORM for model definitions and associations. The database stores user accounts, login logs, the complete academic hierarchy, paper metadata, orders, mock test questions, attempt history and application settings.

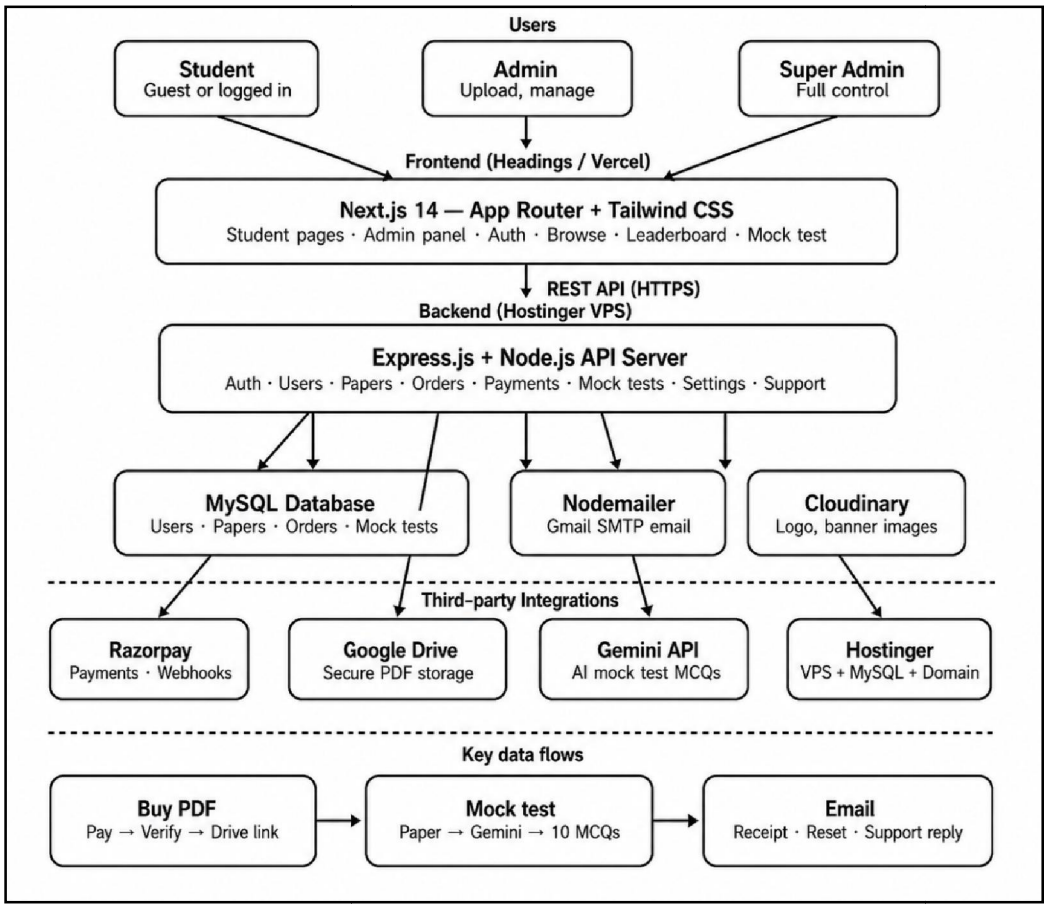

Fig. 1. System Architecture

VII. AI-BASED MOCK TEST WORKFLOW

The Dynamic Mock Test Generation system follows a structured workflow to ensure question quality, diversity and assessment integrity.

Step 1 — Content Ingestion: The super admin uploads question paper text through the admin panel when adding a paper. This text is stored in the papers table as a LONGTEXT field and serves as the source material for AI question generation.

Step 2 — Prompt Engineering: When a student initiates a mock test, the backend constructs a structured prompt for the Gemini 2.5 Flash API. The prompt instructs the model to generate exactly 10 MCQs with four labelled options (A, B, C, D) and one correct answer in strict JSON format, based solely on the provided question paper text.

Step 3 — Response Validation & Storage: The API response is parsed, validated as a JSON array and stored in the mock tests table. This ensures subsequent requests for the same paper retrieve cached questions without additional API calls, reducing latency and API costs.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, May 2026
Impact Factor: 8.2

Step 4 — Attempt Management: Each attempt shuffles the stored questions and options using the Fisher-Yates algorithm, ensuring different question orderings for each of the three allowed attempts. Correct answers are never transmitted to the client; they are compared server-side upon submission.

Step 5 — Scoring & Leaderboard: Upon submission, the backend scores the attempt, stores the result in mock attempts and recalculates the student's overall average. Students who complete five or more mock tests become eligible for the PYQSPOT leaderboard, ranked by average percentage.

VIII. ALGORITHMS USED

A. Fisher-Yates Shuffle Algorithm: Used to randomize the order of questions and answer options for each mock test attempt. The algorithm operates in O(n) time complexity, guaranteeing uniform random permutation of the question array without repetition, ensuring each attempt presents a unique question sequence.

B. HMAC-SHA256 Payment Verification: Razorpay payment signatures are verified using a Hash-based Message Authentication Code with SHA-256. The backend reconstructs the expected signature from the order ID and payment ID using the secret key and compares it with the received signature, preventing payment fraud and replay attacks.

C. JWT Authentication: JSON Web Tokens with a configurable expiry (default 7 days) are used for stateless authentication. The token payload encodes the user ID and role, enabling role-based access control across all protected API endpoints without server-side session storage.

D. Bcrypt Password Hashing: User passwords are hashed using bcrypt with a work factor of 12 salt rounds, providing adaptive resistance against brute-force attacks as computational power increases.

E. Leaderboard Ranking Algorithm: The leaderboard ranks students by computing the arithmetic mean of all mock attempt percentage scores using SQL aggregation (AVG function). Only students with five or more completed attempts are included, ensuring statistical significance in the rankings.

IX. RESULTS & ANALYSIS

A. Platform Adoption: The platform currently serves over 2,000 active SPPU engineering students with a repository of 100+ verified solved PDF solutions covering Mechanical Engineering, Computer Engineering and AI & Data Science (AIDS) branches across multiple semesters and examination patterns (2019 and 2024 patterns).

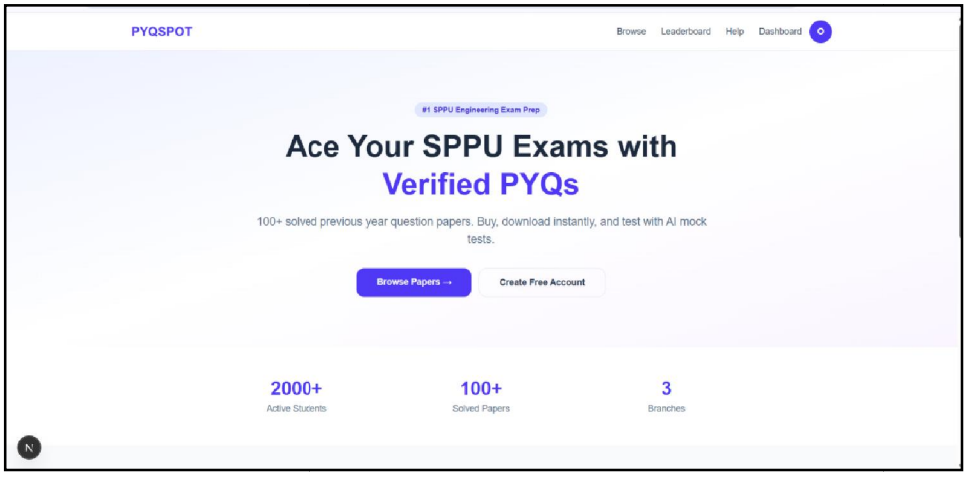

Fig. 2. A screenshot of UI Home page

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, May 2026 Impact Factor: 8.2

B. Payment & Delivery Performance: The Razorpay-integrated payment pipeline achieves a transaction success rate consistent with industry standards. PDF delivery via the Google Drive API is automated and triggered within five seconds of payment verification. All transactions generate automated email receipts with IST timestamps.

C. AI Mock Test Performance: The Google Gemini 2.5 Flash API generates 10 contextually relevant MCQs per request with an average response time suitable for real-time use. The Fisher-Yates shuffling algorithm ensures unique question ordering across all three permitted attempts per paper, maintaining assessment integrity.

D. System Performance: The Next.js SSR architecture delivers pages in under 200ms under normal load conditions. The Express.js backend handles concurrent API requests efficiently through Sequelize connection pooling with a maximum of 10 simultaneous database connections.

E. Admin Efficiency: The dynamic hierarchy management system allows super admins to add new universities, branches, semesters, subjects and papers through the admin dashboard without any code changes, reducing operational overhead significantly.

X. FUTURE SCOPE

Pan-India University Expansion: Adapting the modular Next.js architecture to support syllabus structures from other major Indian universities such as Mumbai University (MU), Pune University (non-engineering) and technical universities across Maharashtra.

Adaptive Learning Engine: Implementing an AI-driven performance analytics system that identifies weak subject areas from mock test history and recommends specific papers for targeted revision.

Video Lecture Integration: Extending the platform to support short-form video explanations linked to specific questions and topics within the PYQ repository.

Push Notifications: Implementing web push notifications to alert students about new paper uploads, exam schedule reminders and performance milestone achievements.

Mobile Application: Developing a React Native mobile application to extend PYQSPOT's reach through iOS and Android platforms with offline paper access capabilities.

AI Difficulty Classification: Using NLP techniques to automatically classify uploaded question papers by difficulty level and topic, enabling filtered browsing by difficulty.

XI. CONCLUSION

PYQSPOT successfully addresses the critical gap in the SPPU engineering examination preparation ecosystem by delivering a secure, scalable and AI-integrated educational platform. By combining a verified PYQ repository with automated digital commerce, real-time AI-powered mock test generation and a comprehensive administrative dashboard, PYQSPOT transforms the examination preparation experience from passive content consumption to active, evaluated and data-driven learning.

The platform's commercial success, demonstrated by over 2,000 active students and a growing repository of 100+ verified solutions, validates both the technical architecture and the market demand for such an integrated solution. The implementation of the Google Gemini 2.5 Flash API for dynamic MCQ generation represents a significant advancement in applying large language models to domain-specific educational assessment, while the secure Razorpay and Google Drive pipeline establishes a reliable template for digital educational commerce.

PYQSPOT demonstrates that modern web technologies, when thoughtfully integrated with artificial intelligence and secure cloud services, can create meaningful, revenue-generating solutions to real academic challenges. The platform's dynamic hierarchy management system ensures long-term scalability, positioning PYQSPOT for expansion to additional universities, branches and examination patterns in future iterations.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, May 2026
Impact Factor: 8.2

REFERENCES

[1]. I. M. Romi, "Adaptive E-Learning Systems Success Model," Journal of Educational Technology, vol. 15, no. 3, pp. 112–128, 2023.

[2]. S. A. Ali, "Designing Secure and Robust E-Commerce Platforms for Public Cloud," Asian Bulletin of Big Data Management, vol. 3, no. 1, pp. 45–62, 2023.

[3]. Deng, X.., He, X., & Zhu, Y. (2024) Innovative Question Bank Management System – Leveraging Digital Technology for Learning. Journal of Digital Innovation for Humanity - JDIH, Vol. 5, pp. 1-13,September 6 2024.

[4]. Jawad Alshboul, Erika Baksa-Varga, " A Hybrid Approach for Automatic Question Generation from Program Codes " (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 15, No. 1, 2024

[5]. K. V. S. Prasanna Vasavi; P. Harshavardhan; K. Pavani; V. Jaya Kanth; K. D. S. R. Manohar; M. Nageswararao. (2025), Online Quiz Management System for Institution." International Journal of Innovative Science and Research Technology (IJISRT) IJISRT25NOV1521 2824-2832. DOI: 10.38124/ijisrt/25nov1521.Next.js Official Documentation, "Server-Side Rendering and API Routes," Vercel Inc., 2024. [Online].

[6]. Razorpay, "Payment Gateway API Documentation," Razorpay Software Pvt. Ltd., 2024. [Online]. Available: https://razorpay.com/docs

[7]. Google, "Gemini API Documentation," Google DeepMind, 2024. [Online]. Available: https://ai.google.dev/docs

[8]. P. O. Kisan, A. S. Pawar, V. B. Satalkar, S. S. Thakare, and R. S. Kanade, "PyqSpot: A Smart System for Engineering Preparation," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 6, no. 1, pp. 187–193, Jan. 2026